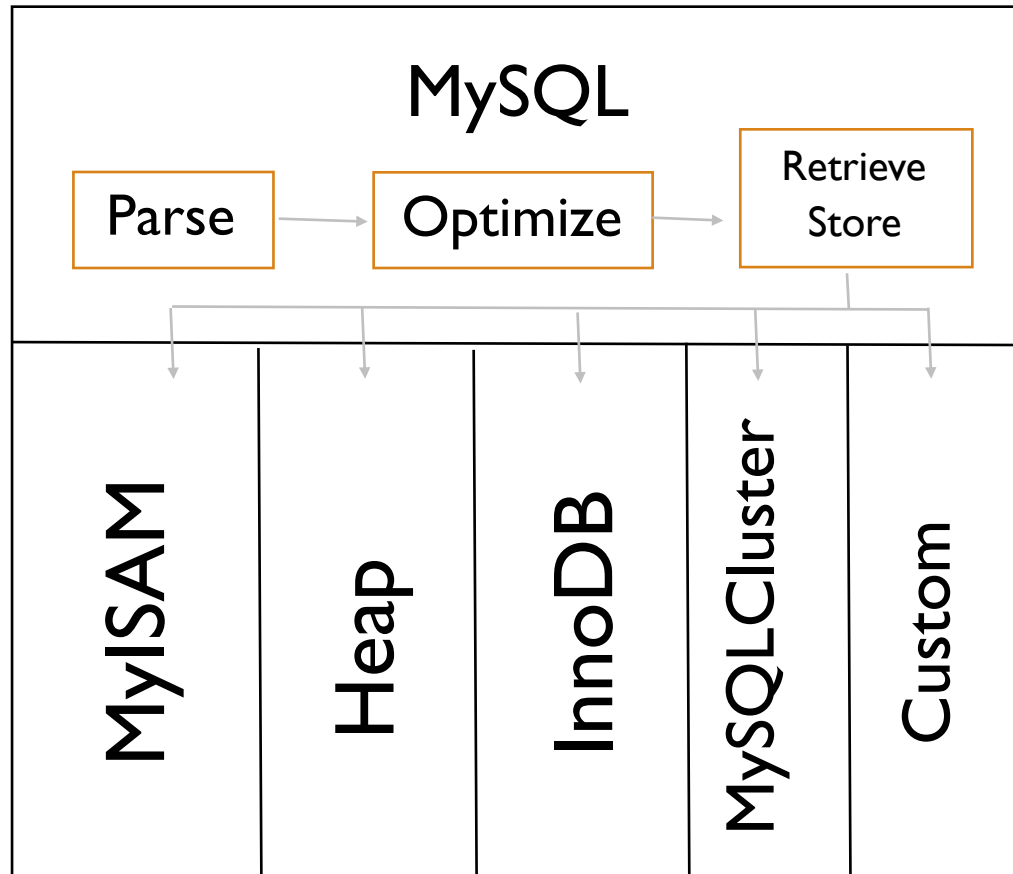


# Understanding the Plugin API

Brian Aker

# Basic Architecture



**MySQL Database  
Management Level**

---

**Table Handler /  
Storage Engine  
Level**

# There are a lot of engines...

- Memory
- Archive
- MyISAM
- Innodb
- SolidDB
- PBXT
- AWS
- ha\_memcache
- myhttp
- Cluster
- Infobright
- Nitro
- Merge
- CSV
- Federated
- ODBCFC
- Falcon
- Blackhole...

# SHOW ENGINES

```
mysql> show engines;
```

Engine	Support	Comment	Transactions	XA	Savepoints
ndbcluster	DISABLED	Clustered, fault-tolerant tables		YES	NO   NO
MRG_MYISAM	YES	Collection of identical MyISAM tables		NO	NO   NO
BLACKHOLE	YES	/dev/null storage engine (anything you write to it disappears)		NO	NO   NO
CSV	YES	CSV storage engine		NO	NO   NO
MEMORY	YES	Hash based, stored in memory, useful for temporary tables		NO	NO   NO
FEDERATED	YES	Federated MySQL storage engine		YES	NO   NO
ARCHIVE	YES	Archive storage engine		NO	NO   NO
InnoDB	YES	Supports transactions, row-level locking, and foreign keys		YES	YES   YES
POSTGRES	YES	Postgres storage engine		NO	NO   NO
MyISAM	DEFAULT	Default engine as of MySQL 3.23 with great performance		NO	NO   NO

# SHOW PLUGINS

```
mysql> SHOW PLUGINS;
```

Name	Status	Type	Library
MEMORY	ACTIVE	STORAGE ENGINE	NULL
MyISAM	ACTIVE	STORAGE ENGINE	NULL
InnoDB	ACTIVE	STORAGE ENGINE	NULL
ARCHIVE	ACTIVE	STORAGE ENGINE	NULL
CSV	ACTIVE	STORAGE ENGINE	NULL
BLACKHOLE	ACTIVE	STORAGE ENGINE	NULL
FEDERATED	ACTIVE	STORAGE ENGINE	NULL
MRG_MYISAM	ACTIVE	STORAGE ENGINE	NULL

# What does a plugin look like?

```
mysql_declare_plugin(memcache)
{
    MYSQL_STORAGE_ENGINE_PLUGIN,
    &memcache_storage_engine,
    "MEMCACHE",
    "Brian Aker, Tangent Org",
    "Simple Interface for working with memcache as a storage engine",
    PLUGIN_LICENSE_GPL,
    memcache_init_func, /* Plugin Init */
    memcache_done_func, /* Plugin Deinit */
    0x0005,
    NULL,                /* status variables */
    NULL,                /* system variables */
    NULL,                /* config options */
},
```

# What about defining features?

- `tables_flags()`
- `index_flags()`
- `handlerton flags`

# Table Object Methods

- `base_ext()`
- `create()`
- `delete_table()`
- `rename_table()`
- `optimize()`, `repair()`, etc...

# DML

- `write_row()`
- `update_row()`
- `delete_row()`
-

# Reading Data

- Scan Read
  - `rnd_init()`
  - `rnd_next()`
- Index Reads
  - `index_read()`
  - `index_next()`

# update\_row()

```
T@3 :||| |>ha_tina::info
T@3 :||| |<ha_tina::info
T@3 :||| |>ha_tina::rnd_init
T@3 :||| |<ha_tina::rnd_init
T@3 :||| |>ha_tina::extra
T@3 :||| |<ha_tina::extra
T@3 :||| |>ha_tina::rnd_next
T@3 :||| |<ha_tina::rnd_next
T@3 :||| |>ha_tina::rnd_next
T@3 :||| |<ha_tina::rnd_next
T@3 :||| |>ha_tina::update_row
T@3 :||| |<ha_tina::update_row
T@3 :||| |>ha_tina::rnd_next
T@3 :||| |<ha_tina::rnd_next
T@3 :||| |>ha_tina::update_row
T@3 :||| |<ha_tina::update_row
T@3 :||| |>ha_tina::rnd_next
T@3 :||| |<ha_tina::rnd_next
T@3 :||| |>ha_tina::extra
T@3 :||| |<ha_tina::extra
T@3 :||| |>ha_tina::rnd_end
T@3 :||| |>ha_tina::write_meta_file
T@3 :||| |<ha_tina::write_meta_file
T@3 :||| |<ha_tina::rnd_end
```



Update occurs here

# How to load an engine?

```
mysql> INSTALL PLUGIN memcache_servers SONAME  
'libmemcache_engine.so';
```

# Web Services Engine

- Generic web service engine.
- SELECT/INSERT/UPDATE/DELETE all mapped to generic web service architecture.
- Combined with new XML features you can parse RSS feeds
- Sister project, mod\_methods, makes it easy to begin integration and testing with sites using Apache.

# Web Service Examples

```
mysql> CREATE TABLE `d` (`a` varchar(125), b text, primary key(a)) ENGINE=HTTP DEFAULT CHARSET=latin1  
CONNECTION="http://en.wikipedia.org/wiki/";
```

```
CREATE TABLE `krow` (  
  `a` varchar(125) NOT NULL DEFAULT "",  
  `b` text,  
  PRIMARY KEY (`a`)  
) ENGINE=HTTP DEFAULT CHARSET=latin1 CONNECTION="http://krow.livejournal.com/data/"
```

```
mysql> select EXTRACTVALUE(b, '/rss/channel/title') from krow WHERE a="rss";
```

```
+-----+  
| EXTRACTVALUE(b, '/rss/channel/title') |  
+-----+  
| Brian "Krow" Aker's Idle Thoughts |  
+-----+  
1 row in set (1.29 sec)
```

# AWS Engine

- Alpha Today
- Allows you to select and store information into Amazon's S3 service

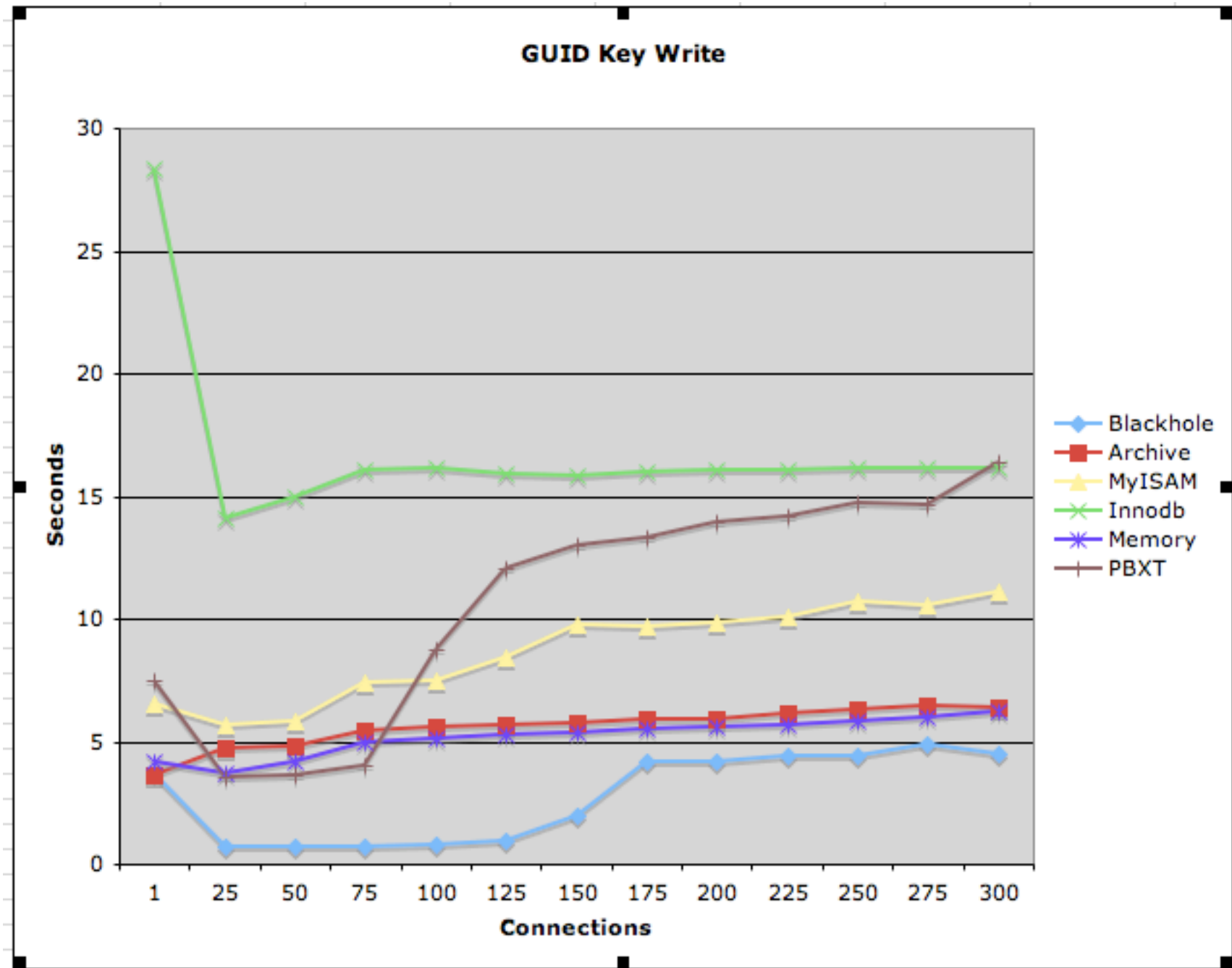
# memcache engine

- Designed to make caching architecture easier
- Can use as many memcached servers as are made available
- Beta today
- Multiple MySQL Servers can share the same data

# PBXT

- Transactional
- Designed for heavy INSERT usage.
- Crash Recovery

# How does it perform?



# A couple of thoughts...

- Start from an example engine.
- Make your engine pluggable from the beginning.
- Automate your testing.
- When in doubt, see how another engine handles the query!

# Resources

- [internals@mysql.com](mailto:internals@mysql.com)
- <http://hg.tangent.org/skeleton-mysql-engine>
- <http://forge.mysql.com/>
- <http://dev.mysql.com/>